# Electronic Regulation of Data Sharing and Processing using Smart Ledger Technologies for Supply-Chain Security

Gregory Epiphaniou, *Member, IEEE,* Prashant Pillai, *Senior Member, IEEE,* Mirko Bottarelli, Haider Al-Khateeb, *Member, IEEE,* Mohammad Hammoudesh, *Senior Member, IEEE,* and Carsten Maple, *Senior Member, IEEE*

***Abstract*—Traditional centralised data storage and processing solutions manifest limitations with regards to overall operational cost and the security and auditability of data. One of the biggest issues with existing solutions is the difficulty of keeping track of who has had access to the data and how the data may have changed over its lifetime; while providing a secure and easy-to-use mechanism to share the data between different users. The ability to electronically regulate data sharing within and across different organisational entities in the supply-chain (SC), is an open issue, that is only addressed partially by existing legal and regulatory compliance frameworks. In this work, we present Cydon, a decentralised data management platform that executes bespoke distributed applications utilising a novel search and retrieve algorithm leveraging metadata attributes. Cydon utilises a smart distributed ledger to offer an immutable audit trail and transaction history for all different levels of data access and modification within a SC and for all data flows within the environment. Results suggest that Cydon provides authorised and fast access to secure distributed data, avoids single points of failure by securely distributing encrypted data across different nodes while maintains an "always-on" chain of custody.**

***Index Terms*—Supply-chain, Blockchain, smart contracts, Hyperledger, Docker.**

## I. Introduction

THE the advent in data collection, generation and storage systems such as the Internet of Things (IoT) and Cloud computing technologies lead to the creation of massive volumes of personal and sensitive data. The value of such data has lead to the proliferation of threats with their implications to domains beyond the electronic space. This phainomenon has not only raised the necessity to secure interactions between all the components within the supply-chain (SC); but equally to develop reliable and cost-effective systems that offer adequate levels of security and authenticity for data to combat fraud and integrity violations [1].

The inability to adaquately control data flows within the SC can impose significant security risks that can manifest adverse impact on data restoration processes and critical business operations [2]. The global SC currently faces a significant issue to distinguish counterfeit parts from legitimate ones as the technology used by organised criminal groups (OCGs) has improved significantly [3]. In addition, several challenges exist to regulate electronically data sharing processes between third party suppliers and external entities while maintaining clear audit trail of data related activities [4]. One partial solution to this problem, is often to deploy third party auditors (TPAs) to verify data integrity and reliability between all entities sharing data within the SC [5]. However, conventional and often scheduled SC audits are often unable detect anomalies and handle deficiencies related to data misuse leading to issues such as erroneous demand forecasting and incomplete operations planning [6].

The geographical dispersion of SC facilities can also influence the decisions related to which data related activities should be performed in each facility [7]. The decision is often based on the efficiency of centralising versus decentralising these activities as a function of the proximity to customers and suppliers. Decision-making based on proximity has usually strong impact on the cost and performance of the SC where data management services play a key role in defining the trust requirements on third-party auditors. The extent to which the data is accurate, timely and complete can influence decision-making regarding other parts of the SC such as coordinating daily activities and forecasting and planning to anticipate future demands. The decision on how much information to share with other companies and how the integrity and authenticity of that data is assured online as a business enabler is a balancing act between openness and responsiveness linked to profitability to fragmented and fast moving markets [8].

Elements such as outsourcing, partnering, and in-house expertise, are considered key performance metrics of the core SC operation within which data sharing processes are often manifested to better understand the requirements of the customers and define and develop SC capabilities further. Data is a strategic business resource that dictates the deployment of suitable sharing models that can offer traceability, integrity and security aspects throughout data's whole life-cycle. The use of Blockchain (BC) technology as a distributed platform for data management purposes has slowly emerged in the literature as a way to not only support transaction execution in several SC processes, but equally to verify the correctness and traceability of data [9], [10], [11].

At high level, BC is a distributed ledger that its accuracy,

G. Epiphaniou, P. Pillai, M. Bottarelli and H. Al-Khateeb are with the Wolverhmaption Cyber Research Institute (WCRI), School of Mathematics and Computer Science, University of Wolverhampton, West Middlands, UK

M. Hammoudeh is with the School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester, M15 6BH, UK

C. Maple is Warwick Manufacturing Group (WMG), University of Warwick, International Manufacturing Centre, Coventry, UK

Manuscript received April 19, 2005; revised August 26, 2015.

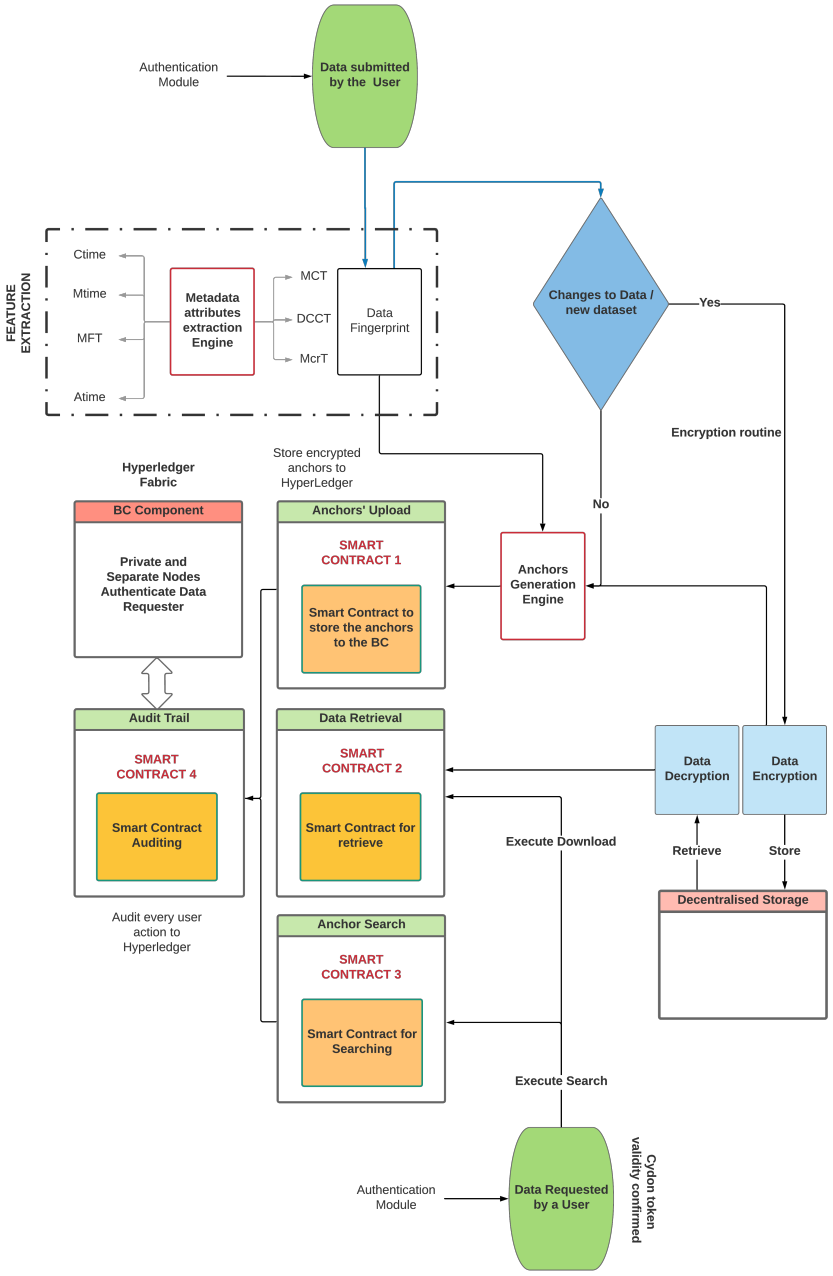



Fig. 1: Cydon Platform

validity and consistency can be verified by the use of dedicated consensus mechanisms in a distributed fashion [12]. Strong cryptographic primitives are usually deployed to assure the immutability of the ledger with a variety of implementations, trust models and threat models as appropriate [13]. In these environments data integrity and immutability can be considered as key security components irrespective of the IT infrastructure used such as Cloud service delivery models, storage server technologies and architectures [14]. These technologies act as trusted third parties to verify the reliability and accurateness of the data transmitted, stored and processed and seek to rather centralise data processing which makes them susceptible to a number of issues around data availability, integrity and high operational costs [15].

The impact of security breaches in SC dictates a clear incident management approach where detection, prevention, response and mitigation doctrines are at its core [16]. The threat landscape in SC is not only constantly evolving but also presents unique risks associated to disruption of services while the co-ordination of organised cyber crime rings makes them harder to detect. There are several standards and procedures introduced in the public domain with regards to SC management but the efficacy of these practices remain largely incomplete due to the heterogeneity of responses and the different ways that risk is assessed both internally and externally to these environments [17]. Elements such as dependencies between exporters and importers, logistics, ship-

ping management, insurance and government bodies create a complex and unique SC ecosystem in terms of the security requirements that poses significant challenges in maintaining the chain of custody [18]. The adaptation of BC technologies can influence metrics in Supply-chain Management (SCM), such as transportation cost modelling and optimise existing warehouse efficiency models. These technologies are promised to decompose sources of information and suppliers' relationships better. The different ways that Distributed ledger technologies (DLT) can be deployed can create value for the supply chain manager with regards to all scheduling activities related to manufacturing, testing, packaging and preparation for delivery [19]. This can hence improve levels of production output and overall productivity from all differences within the SC while optimising specific processes within the logistics and information management systems [20]. Authors in [21] emphasise on perceptual economic models based on BC as a means to better articulate specific layers of value creation recording and actualisation. The exponential increase in distributed applications has revolutionised the SC optimisation tools by optimally allocating capital resources and business growth processes and data management [22], [23].

The remainder of this paper is structured as follows: Section II discusses related works in the field of distributed ledger technology with focus on its application to data processing and management. Section III presents the design, implementation and testing of our Cydon data management platform and its novel metadata search and retrieve algorithm. The proposed solution consists of an underlying Private Permissioned Blockchain and a distributed storage technology coupled with a novel algorithm for anchor generation used for a secure and "off-chain" data search and retrieval. The execution of our algorithm is done by the development and deployment of bespoke smart contracts directly injected in the Blockchain network. In Section IV, we present a preliminary threat model against our solution with emphasis placed upon our dedicated smart contracts. Initial results of Cydon's performance and testbed parameters are discussed in Section V. Finally, Section VI concludes our work.

## II. Related Works

DLTs are gaining momentum in the last few years as core technologies that they can offer significant improvements to existing data-processing capabilities [24]. The new immutability and transparency aspects that BC technologies can offer are promised to minimise human error increase the overall trust to all business processes and optimise specific data governance processes [25]. Typically, a BC network consists of many nodes that maintain a set of shared states that are frequently changed by associated transactions. The transactional modelling the BC utilises does not differ significantly from the traditional database model other than the assumption that the nodes in a BC network may exhibit Byzantine behaviour [26].

BC transactions are encapsulated into blocks forming an immutable and indisputable chain using a Merkle tree structure (See Fig. 2). Many cryptographic schemes deploy Merkle trees that establish specific relationships between a tree leaf value

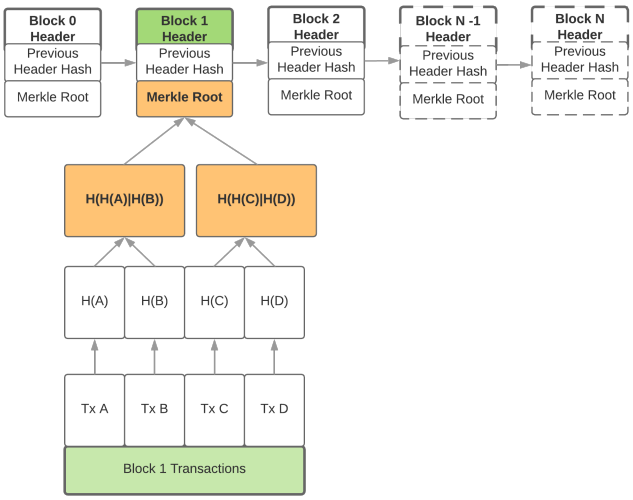



Fig. 2: Blocks of transactions using Merkle trees

and the root node value so as the authenticity of the latter can be established. Sibling leaves are combined and hashed to form a parent leaf repetitively. The traversal mechanism developed allows the values from all leaves to be stored outside the memory space [27]. Depending on the way that these transactions are verified, a distinction between private and public BC deployments is made. The necessity to centrally verify transactions is removed with invalid transactions not be admitted in the BC network while maintaining some degree of anonymity and auditability in the process [28].

Common assumptions made in private and public BC deployments are: 1) the necessity to store a state and that 2) multiple participants are willing to commit transactions. In cases that not all participants are known, a permissionless BC model often applies; whereas if all participants are known but not trusted and public verifiability of transactions is required then a public permissioned BC model is more appropriate. In cases that public verifiability of transactions is not required between untrusted participants then a private permissioned BC is often deployed [29]. In private BC deployments, business logic can be expressed by the development of smart contracts. These smart contracts are often described as distributed applications built on top of the ledger and used to capture key data abstractions that represent key business processes in the form of verifiable transactions agreed by all parties through a consensus [30]. There is evidence to suggest that this is a desirable property useful to sectors such as finance and data management [31],[32],[33],[34].

In terms of using BC technologies for data distribution and processing processes, work in [35] introduced the concept of searching mechanisms over encrypted keywords. The authors have defined a ZKP authentication module for the access to a private permissioned BC with and off-chain storage capability. Although the model promises to introduce certain benefits in terms of performance, no real implementation of the platform is given. Authors in [26], introduced a dedicated BC benchmarking solution to evaluate the performance of existing

technologies against specific data processing and transactional processes. They argue that BC is still immature as a technology to replace traditional database systems, due to the fact that consensus protocols must be optimised further in terms of their increased verification times and admission processing overheads. They do, however, record the improvements made from decoupling the data storage from the actual execution engine by outsourcing the consensus process in existing data models. The optimisation issues of consensus mechanisms have been recently addressed in [36],[37].

The issue of tracing data in a transparent way has been discussed in [38]. The authors have developed a data sharing scheme that seperates data from transactions and enables a logical double chain structure to assure data traceability. This is done by dividing data into blocks that form the basis for the sharing and tracing capabilities developed as part of their scheme using the Practical Byzantine Fault Tolerance (PBFT) consensus protocol. A distinction is made between data that is frequently accessed but infrequently modified in [39]. The authors introduced a framework that combines Ethereum and Hadoop distributed filesystem (HDFS) for public BC deployments. Data retention of large amount of data for longer periods is of paramount importance and a driving factors behind this work. The framework assures certain degree of history tracking in off-chain storage systems in a decentralised fashion using persistent storage of forensically relevant evidence (metadata). Their results suggest a quicker response time of reading and validating a file almost inversely proportional to its size. This research area was also explored in [40] where homomorphic verifiable tags were used to assure provable data possession (PDP) using dynamic data while virtualised machines executed the smart contracts on the BC. The integrity features on this work were identified as well-formed transactions, separation of duty, authentication, audit, principle of least privilege, objective control and control over privilege transfer, following the Clark-Wilson model. The issue of data retention and secure log distribution and sharing was also investigated for deployments in existing Cloud and associated service delivery models (IaaS, PaaS, SaaS) [41] . The authors have emphasised on the preservation time of logs and its importance to threat identification arguing on the scalability of centralised solutions when dealing with large scale data logs. A private BC has been proposed as a storage log management system across different organisational entities and devices using a novel checkpoint logs that guarantee the integrity of all logs based on their number and fixed timeframes. In terms of Cloud deployments, works in [42], [43] uses the Cloud as the main storage facility where the BC network is used as a controllable trustworthy data management solution where voting and vetting on data validity is controlled by a trust authority. The BC network becomes the voting station that authorises (or not) further changes to data by distinguishing between voting and counting distributed applications executed on the BC.

SC collaboration is an acknowledged and significant challenge within the Pharmaceutical industry [44]. Sharing of information and controlling intellectual property are a concern across two specific activities: 1) the co-development of new substances/products (drugs) and 2) planning activity external to the organisation. A typical drug development programme estimates to have all its medical records electronically stored and processed [45]. There is a wide range of skills required to translate research into commercially available drugs. As a consequence, collaborative arrangements between organisations are common as a means of exploiting organisational competencies and reducing financial risk. It is therefore clear that within the Pharmaceutical SC, BC technology can offer the verification of the intellectual contribution made by collaborative parties and the secure sharing of sensitive intellectual property and regulated data with clinics, regulators, physicians and partners such as charities and Contract Research Organisations [46].

The key assets created during a pharmaceutical development programme are: 1) the formulation (GMP standard), 2) the data package and 3) the associated intellectual property rights. Where more than one organisations contribute towards the creation of these assets, it is important to keep a record of respective contributions rendering BC technology a suitable candidate to this domain. The sharing of patient-level data, following clinical trials, is viewed as being important to ongoing scientific research that has the potential to deliver benefits to future patients and society [47]. There are certain requirements to maintain privacy of clinical trial participants which has led to the following outcomes: 1) the data holder must anonymise or de-identify the data, before sharing it with researchers [48]. This creates problems in so much that the data becomes less useful for answering scientific questions and it increases the chance that misleading interpretations of the data might occur. 2) Over-reliance on legally binding data sharing agreements (between data holder and researcher) that define the usage limitations on the data. 3) A call for controlled access of trial data via a secure locked box system.

Distributed Cloud storage architectures based on BC have been introduced in the literature to address e2e security issues. [49]. In their work, the authors distinguish between users who need cloud storage service and users that can supply empty storage as part of their unstructured P2P environment. An optimisation scheme is developed to solve the file block replica placement issue between all entities that hold multiple copies of the data. Results suggest no loss of data across the whole infrastructure while reducing the overall transmission delay in comparison to alternative data centre architecture(s).

A publicly verifiable data deletion protocol has been introduced in [50]. The authors use a dedicated timestamp server as a the trusted authority to provide the necessary information to the cloud server for the production of the data deletion proof. The existing threat model assumes a “semi-honest" server as part of their analysis but more is needed with regards to the possible clock de-synchronisation attacks against the protocol.

The concept of secure and private data sharing using DLTs has been explored in [51]. The main research focus is to provide privacy while maintaining fine-grained access control over the data shared. Recent legal and regulatory requirements for entities possessing and sharing data dictate appropriate data verification mechanisms in place such as data custodians, rendering the process as time-consuming and often inefficient.

The complete lack of sharing facilities for data custodians has propelled research in personal data storage (PDS) solutions, especially in cases that different types of data need to be aggregated and appropriate control to be attributed to the data owners [52],[53].

## III. Cydon Data Management Platform

Cydon is a reliable and functional data management platform that runs over a dedicated cluster of smart contracts through a web-portal. This platform creates transactions based on relevant logs produced using a search and retrieve algorithm executed on the target network by invoking multiple smart contracts (See Sec. III-C). These logs are cryptographically hashed and encapsulated into blocks and pushed into a private permissioned BC for verification purposes using Hyperledger Fabric. This is achieved with the development of relevant distributed application(s) (smart contracts) running by our platform. The actual data is distributed and stored securely across multiple locations using an off-chain storage capability to ensure data availability and redundancy inside the network. Authorised users access the system, logs and data via a dedicated web portal.

The overall structure of our Cydon platform is illustrated in Fig. 1. The platform utilises a metadata attribute extraction engine where forensically relevant information is extracted for any given data. There are seven attributes extracted and used as an input to generate a unique fingerprint for each data. That fingerprint is then used as a keyword (anchor) that is stored on the blockchain network via a transaction executed through a dedicated smart contract. All anchors are stored in an encrypted fashion using a set of strong iterative block ciphers via the users' GUI. To upload a new file, the user authenticates himself in the frontend and sends a file to a specific endpoint using a convenient web form. The file is first encoded using 256bit AES whose initialisation random vector is inserted at the beginning of the file. The resulting stream is then forwarded to IPFS to be stored and to retrieve the corresponding hash. The latter is the first component of a data structure that also includes any keywords provided by the user and all metadata extracted from the file. Among them, we mention the original name of the file, its creation and modification date, the size and the owner. The metadata is serialised as a single buffer of bytes which is encoded using the private key of the Cydon token; hence, it is accessed as a single and indivisible entity. On the other hand, the keywords are individually encoded and saved in an array allowing the smart contract to examine the relevance of the results without decoding them. Finally, the entire structure is serialised and stored as the blockchain transaction payload. The key associated with the latter coincides with the public id of the Cydon Token, which will become an index in the state-database underlying each peer, making search queries almost instantaneous. The requester can traverse the BC and execute searches for keywords (anchors) that correspond to the location of encrypted data. The requester executes dedicated search smart contracts associated with both the request and retrieval processes. These processes are encapsulated and mapped against a specific data object for auditing and validation purposes, namely our Cydon token explained in Sec. III-B. All attempts to upload, search and retrieve or manipulate data are recorded for each user and associated tokens. There is an audit distributed application that automatically creates a report of actions and transactions carried out for all tokens in the platform for auditing and compliance purposes. The encryption and decryption processes are executed on-the-fly by probing the location of encrypted data within an IPFS distributed storage network across all nodes within the Private permissioned BC environment. IPFS is a combination of a P2P network and a protocol for data storing in a distributed manner using content-addressing to identify each file in a global namespace. Files are distributed using a BitTorrent-based protocol creating links between nodes that store the content using cryptographic hashes. It utilises the Merkle Directed Acyclic Graphs data structure as part of its core operation. This enables IPFS to offer content addressing, de-duplication and tamper-proof capabilities in terms of the way that data is stored and indexed in our Cydon platform [54].

### A. Design and Requirements

Sharing sensitive information with third parties who are not part of a formal collaboration does occur within several SC. For instance, charities are a significant source of research funding, along with Government departments, and there is an expectation that representatives of these organisations will be kept informed of progress. Giving updates to third parties provides ample opportunity for knowledge / information leakage. While inadvertent disclosure of data is a problem, the high commercial value of the data being exchanged also makes it a target for organises cyber-crime rings. For example, drug development companies that rely upon Contract Research Organisations (CRO) to perform elements of their research programme required to share sensitive information so that the planned research can be specified correctly. Whenever there is sharing of sensitive information there is a need for suitable technology that minimises the capacity of other parties to disclose it. Our Cydon platform provides a secure information sharing environment and allows the identification of any individual who accesses data to be established. The nature of our BC also ensures that any unauthorised changes to data can be pinpointed and attributed to a given token after the authorisation process is completed. The assumption here is that all data flows are passing through our Cydon platform. Specific APIs have been developed to provide interoperability between Cydon and de-facto authentication schemes currently used (E.g. Kerberos, Active directory, SSO, etc). In an attempt to identify the key metrics to be developed as part of our platform the following design requirements have been identified in Table II.

### B. Cydon Cryptographic token

The smart contract operation executes both upload and searching capabilities with completely different APIs developed. The Cydon Token, $Ct$ (See Table III) is a compound object stored in the local database and uniquely identified by

TABLE I: Cydon design requirements

| Design Requirements: | Usage |
|---|---|
| Authentication module | Appropriate authorisation levels to data uploader / requester |
| Anchor generation engine | Metadata extraction attributes to be used as the seeds |
| Anchor linking to the BC | The transaction for the uploader and keyword storage, and the retrieval capability with fast execution |
| Identification of data location | Universal hash matching as part of the searching process and integrity verification |
| Distributed storage location | A customised distributed FS solution to be linked to the smart contract operations |
| Encrypted storage and keygen | Confidentiality aspects as part of the transaction that control uploads and decryption processes from the requester |

TABLE II: Metadata Feature Extraction Attributes

| Features: | Description |
|---|---|
| Ctime | Creation Time: The time the file created |
| Mtime | Modified Time: This timestamp describes when was the last time the file was modified |
| MFT | Master File Table: describes information about a file, including its size, time and date stamps, permissions, and data content |
| Atime | Accessed Time: A timestamp to describe when was the last time the file was accessed |
| MCT | Metadata Change Time: Time the MFT Record was last modified |
| DCCT | Data Content Change Time: Time Data content of a file was last modified |

its primary key (*field Id*), autogenerated after the insertion. The creation of such a token takes place during the upload of new content: the user is able to provide a human readable alias (*field Alias*) and the system automatically generates all other fields, more specifically the symmetric encryption key (field DataKey) used to encode the payload, the salt (field Keyword-Key) used to hash possibly provided keywords and the token's public id. The newly generated token is then associated with a single user through the foreign key UserId, identifying him as the owner. Moreover, it is also linked to many possible recipients through the many-to-many relationship "Permission" which ensures that only allowed users are able to retrieve the original content. The public id is the only part of the token which is stored in the BC's transactions and it can be used to search for the corresponding material. Furthermore, the *public id* (and the Cydon token itself) may be re-used in several uploads, acting as a grouping mechanism to keep different contents together even if submitted in different time instants. In this case we can provide access to specific data associated with muptiple uploads based on the authorisation levels.

### C. Smart Contracts design and Execution

The Cydon system currently includes four smart contracts which are injected in the BC network namely *Upload, Query, download and Audit.* The *Upload Smart Contract,* is invoked at the end of the upload process to generate the final transaction. Assuming the user needs to upload new content, the first step is to access the front-end through authentication. In this case, the new content requires a new Cydon token which can be generated by calling a server-side service through the exposed APIs. Access to the platform is protected by form-based authentication, where users log in using their usernames and passwords. These credentials are sent to an API endpoint that generates a JSON Web Token (JWT) containing the unique identifier of the user, his access level (either admin or regular user) with the validity of one day. The payload is signed via a secret key available only to the server, in order to block any possible infringement. Except for the authentication function, all other APIs require the passage of this token in the call header. In this way, identity management is delegated to the client without requiring the presence of a server-side session, which would limit the scalability of the platform.

Recipients selected and their authorisation level will be stored as token's permissions in the server database. Even if the newly generated token contains the encryption keys for both payload and keywords, only its *public id* is revealed to the client as it is the only part that needs to be shared. At this point, the user can select files for upload and eventually inserts the associated keywords. Files' payloads are sent to the backend where they are encrypted using the chosen token's symmetric key and then stored in the distributed file system which, in turns, returns the *file id* that uniquely identifies the resource. The *public id*, the *file id* and keywords are the fundamental elements for creating an upload-transaction in the BC. The request arrives at the backend again through an exposed API and it is forwarded to the Hyperledger infrastructure where it is then executed. The, transaction's parameters are checked, and the structure is validated against a default schema. If all checks pass, the transaction is created and the world-state db is accordingly updated, completing the upload process. This will allow the allocation of groups of entities to a single token for both the retrieval and modification processes associated with files.

When the user wants to retrieve the file-list associated with a token, he provides to the frontend the public token-id and eventually any keywords needed to restrict the results. Since in this phase the user is already authenticated, the frontend is able to lookup the database-id that uniquely identifies him and verifies that he has at least read access. This check is performed by traversing a one-to-many table linked to the main tokens-table, which contains all the authorised users together with their access level. Even if the user created that token in the first place, his access must be explicitly allowed in the permission table. In this way, the system can preclude writing, and even reading content previously uploaded after being revoked (for example, after a change of duty). The query attempt, its outcome (either allowed or forbidden) and the eventual results are stored in the blockchain for auditing purpose. The download process uses the same token-validation method: besides the desired token, the user also has to provide the hash of the inquired file. Assuming read access is granted

TABLE III: Cydon-token $C_t$ fields and corresponding contexts of use (UP=upload, QU=query, AU=audit, DN=download)

| Field | Type | Usage | Notes |
|---|---|---|---|
| PublicId $C_P$ | guid | UP, QU, DN, AU | Shareable public uniq. id |
| DataKey $C_D$ | byte[32] | UP, DN | Symmetric key (payload) |
| KeywordKey $C_K$ | byte[64] | UP, QU | HMAC key (keywords) |

to the requester, the data is retrieved from IPFS and its integrity is verified by recomputing its hash-value. The file is finally sent to the user and a detailed download transaction is added to the blockchain.

### D. Upload smart contract

The user $U$ starts the upload process by providing a public id $C_P$, a file $F$ and an array of keywords $\underline{K}$ as follows:

1: **if** not $HasWriteAccess(U, C_P)$ **then**
2: return ▷ exit if user has no right to add files to this token
3: **end if**
4: $C \leftarrow Lookup(C_P)$ ▷ get token from database
5: $I_V \leftarrow RandomBytes(16)$ ▷ 128 bits initialisation vector
6: $E \leftarrow AES_e(C_D, I_V)$ ▷ generate AES 256 bits encoder
7: $F_E \leftarrow Concat(I_V, E(F))$ ▷ encode the file
8: $H \leftarrow StoreAndGetHash(F_E)$ ▷ store encrypted file and get hash
9: $M \leftarrow \{F_{name}, U\}$ ▷ store filename and user in metadata
10: $I_V \leftarrow RandomBytes(16)$ ▷ refresh initialisation vector
11: $E \leftarrow AES_e(C_D, I_V)$ ▷ refresh encoder
12: $M_E \leftarrow Concat(I_V, E(Serialise(M)))$ ▷ serialise and encode metadata
13: $\underline{K}_E \leftarrow \emptyset$ ▷ prepare encrypted keywords array
14: **for all** $k \in \underline{K}$ **do** ▷ for all keywords
15: $\underline{K}_E \leftarrow \underline{K}_E \cup HmacSha256(k, C_K)$ ▷ encrypt and append
16: **end for**
17: $StoreUploadTransaction(C_P, H, M_E, \underline{K}_E)$
18: **return**

### E. Query smart contract

The second smart contract needed is the search functionality. In this scenario, the users need only the *public id* of the content they want to retrieve. However, they can also provide a list of keywords to refine the search further. This will enable selective access to files from a list of files associated with a token and authorised to access. Before reaching the BC, the query is first evaluated in the backend to check if the user has access to the Cydon token pointed by the *public id*. The query smart contract is then executed: after the mandatory check of parameters, the chain-code integrates over all upload transactions looking up for the given *public id*. All valid entries are returned to the backend with associated scores and calculated as the number of matching keywords in assisting the user to locate relevant material. Finally, the user can download a specific content, automatically retrieve it with the help of the corresponding *file-id* and then decrypted by the previously acquired Cydon token. The user $U$ starts the query process by providing a public id $C_P$ and an array of keywords $\underline{K}$. The whole execution steps are given as follows:

1: $allowed \leftarrow HasReadAccess(U, C_P)$ ▷ check if user has read access
2: $C \leftarrow Lookup(C_P)$ ▷ get token from database
3: $\underline{K}_E \leftarrow \emptyset$ ▷ prepare encrypted keywords array
4: **for all** $k \in \underline{K}$ **do** ▷ for all keywords
5: $\underline{K}_E \leftarrow \underline{K}_E \cup HmacSha256(k, C_K)$ ▷ encrypt and append
6: **end for**
7: $res \leftarrow \emptyset$ ▷ prepare query result
8: **if** $allowed$ **then**
9: $\underline{T_U} \leftarrow \{t_x \in Blockchain : type(t_x) = upload \wedge t_x.C_P = C_P\}$
10: **for all** $t_u \in \underline{T_U}$ **do** ▷ for all matching uploads transactions
11: $t_u.Score \leftarrow |t_u.\underline{K}_E \cap \underline{K}_E|$ ▷ compute score
12: **end for**
13: $res \leftarrow \underline{T_U}$
14: **end if**
15: $M \leftarrow \{\underline{K}, U_{id}, allowed, res\}$ ▷ store search in metadata
16: $I_V \leftarrow RandomBytes(16)$ ▷ 128 bits initialisation vector
17: $E \leftarrow AES_e(C_D, I_V)$ ▷ generate AES 256 bits encoder
18: $M_E \leftarrow Concat(I_V, E(Serialise(M)))$ ▷ serialise and encode metadata
19: $StoreQueryTransaction(C_P, M_E)$
20: **return** $res$

### F. Download smart contract

The user $U$ starts the download process by providing a public id $C_P$ and the hash $H$ of the file he wants to download as follows:

1: $allowed \leftarrow HasReadAccess(U, C_P)$ ▷ check if user has read access
2: $C \leftarrow Lookup(C_P)$ ▷ get token from database
3: $M \leftarrow \{H, U_{id}, allowed\}$ ▷ store download in metadata
4: $I_V \leftarrow RandomBytes(16)$ ▷ 128 bits initialisation vector
5: $E \leftarrow AES_e(C_D, I_V)$ ▷ generate AES 256 bits encoder
6: $M_E \leftarrow Concat(I_V, E(Serialise(M)))$ ▷ serialise and encode metadata
7: $StoreDownloadTransaction(C_P, M_E)$
8: **if** $allowed$ **then**
9: $F_E \leftarrow GetFileFromHash(H)$ ▷ get encrypted file from storage
10: $I_V \leftarrow F_E[1:16]$ ▷ get initialisation vector from the file head
11: $D \leftarrow AES_d(C_D, I_V)$ ▷ generate AES 256 bits decoder
12: $F \leftarrow D(F_E[17:end])$ ▷ decode the file tail
13: **return** $F$

14: **else**
15: **return** $nil$
16: **end if**

### G. Audit smart contract

The audit can be generated by users with administrative access to the platform by providing the public id $C_P$. This takes the form of a detailed report generation with a clear audit trail of all transactions executed and associated with a specific Cydon token $C_t$. The execution sequences of our audit smart contract are given below:

1: $C \leftarrow Lookup(C_P)$ ▷ get token from database
2: $\underline{T_U} \leftarrow \{t_x \in Blockchain : t_x.C_P = C_P\}$
3: **for all** $t_u \in \underline{T_U}$ **do** ▷ for all matching transactions, of any type
4: $I_V \leftarrow t_u.M_E[1:16]$ ▷ get initialisation vector from metadata head
5: $D \leftarrow AES_d(C_D, I_V)$ ▷ generate AES 256 bits decoder
6: $t_u.M \leftarrow D(t_u.M_E[17:end])$ ▷ decode transaction metadata
7: **end for**
8: **return** $\underline{T_U}$

A core element of Cydon platform is the audit smart-contract which provides an immutable trace of every user action, storing both the requests and the results for all dedicated BC transactions. In fact, each Cydon smart-contract contributes to the generation of the audit by appending context-specific information. For example, the upload and the download smart-contracts write transactions that contain the actual user and the related anchor/file hash.

During the search action the complete search query is stored including the keywords, the corresponding results and a flag to indicate if the user was allowed to execute such search based on the Cydon-token permissions. This permits the creation of an invaluable chronological complete report with both valid and invalid accesses to specific protected content.

## IV. Threat Model

Despite the innovative solutions BC promises to bring to the data decentralisation, storage and processing capability, the technology itself seems to present some inherent security risks. The Cydon system must maintain the key security principles represented by the CIA Triad; confidentiality, integrity and availability. Furthermore, other attributes include Productivity and Proprietary. Table IV summarises potential threats affecting these attributes. We have identified that Spoofing could occur at the network level (e.g. ARP poisoning/spoofing) to facilitate a Man-in-the-middle (MITM) attack against the system where a session could also be hijacked. It could also occur at the application level e.g. when an authentication mechanism is bypassed, and the attacker gain access to the permissioned BC with a spoofed ID.

The system includes a web interface which introduces threats by means of Cross Site Scripting (XSS) and SQL injection. Furthermore, gaining access to the network could allow tampering to data in transit. This type of data is not yet hashed into the BC and the system would therefore be vulnerable to such attack. The BC provides means to maintain and verify records' integrity. However, data in transit will go through encryption tunnels using protocols such as TLS and signed Certificates to thwart tampering attempts.

Information leakage will occur if access control fails; an attacker could consider gaining a copy of the SQLite DB through an injection attack to brute-force the user credentials of sys-admins. Information can also be leaked at the user end if their system is compromised (e.g. keylogger software). Furthermore, any unpatched vulnerability (e.g. zero-day) in the infrastructure and software installed could allow data exfiltration from the network. The system includes a distributed ledger which is good to resist Denial of Service (DoS) due to floods of requests. However, other parts such as the Web Service is central and could be overwhelmed with a DDoS. Backup, various levels of data redundancy, and fall-back are all means to adapt to DoS attacks where possible. Since Cydon provides multiple copies of encrypted data stored in a distributed fashion, it does offer certain level of protection in terms of data redundancy.

An implementation vulnerability or misconfiguration (e.g. weak admin password) could allow an attacker to escalate privilege in the system. In response, authorisation will be enforced with access to resources facilitated via access control systems. Other threats include system misconfiguration and unpatched products. This can be hardware such as Internet gateways, or software such as a Web interface vulnerable to SQL injection attacks. Nonetheless, in permissioned BC implementations, only pre-selected set of trusted nodes can create new transactions. Therefore, there is a threat that an attacker exploits this established trust model by gaining access to one or more of these nodes and disrupts the system at large.

Attacks such as the devaluation of cryptocurrencies, loss of mining rewards, or even closure of cryptocurrency exchanges do not apply in our platform since there is no mining process taking place or crypto-wallet existent within the network. The private nature of the implementation renders specific attacks applicable to open blockchain networks such as selfish mining more challenging to implement against Cydon. Since all entities are authenticated to the platform database, security and associated controls have been implemented to mitigate some of the security risks. Although a systematic decomposition of our smart contracts' data-flow is outside the scope of this work, no initial patterns were identified to imply violation of the security principles and properties assumed during their development. Part of our future work is to audit all smart contracts developed using the tool introduced in [55]. The authors propose a smart contract auditing approach were the data-flaw of smart contract in Ethereum is symbolically encoded. The solution promises to automatically check the compliance of the smart contract against security patterns in an attempt to prove or disprove functional security properties.

To create transactions or to read their contents, the user must be authenticated in the application frontend, so that it is possible to extract his blockchain access credentials subsequently verified by the Certificate Authority (CA) of the organisation to which he belongs. During the invocation

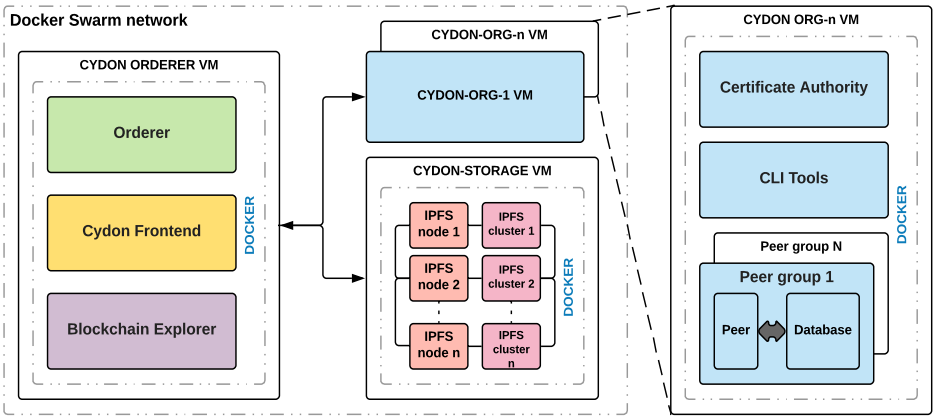
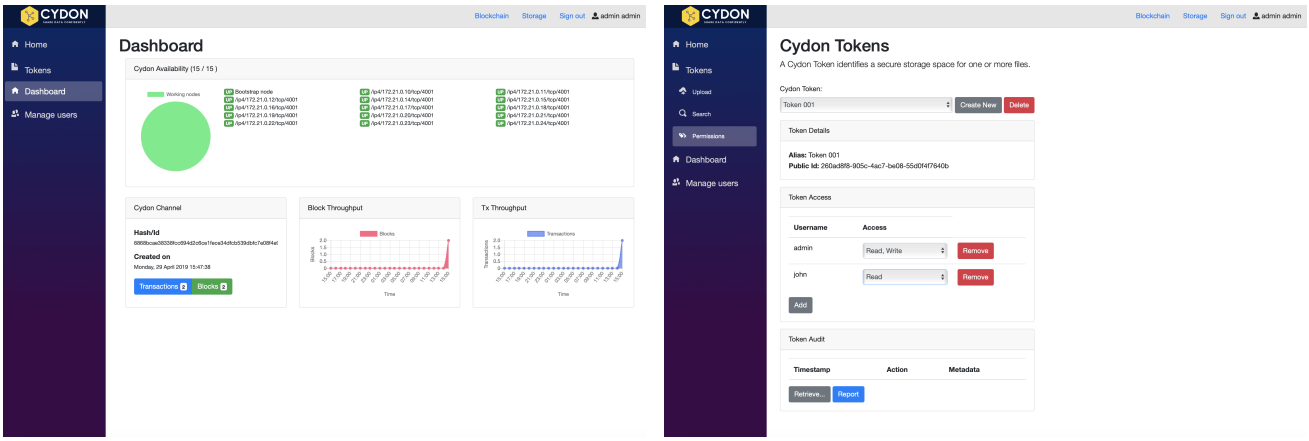



Fig. 3: Cydon Testbed




(a) Cydon Dashboard (b) $Ct$ Permissions

Fig. 4: Cydon's Frontend Container for the dashboard and $Ct$ permissions

of smart contracts, our approach is strengthened by adding further structural checks. We achieve that by comparing the transactional payload with a strict JSON scheme published in [56]. We refuse execution if properties deemed mandatory are missing or non-compliant. Moreover, the existence of unnecessary properties is also considered a blocking factor, both to avoid unwanted data and to limit the growth of the Blockchain undermining its scalability.

Currently, there is a clear trade-off between the block size and the ability of the BC network to resist against possible DDoS attacks. Bigger blocks allow more records to be stored at the expense of complicating the running and managing processes in the BC nodes. The distributed data storage capability creates an increased attack surface that can provide a skilful adversary with alternative ways to access data stored at the nodes. Data mining, data correlation and traffic analysis can be utilised by skilful adversaries to retrieve valuable information related to smart contracts, users, network structure and applications running. Issues associated with traditional Public key cryptography are also manifested with regards to the private key management, confirmation of the user's identity and assumptions about key usage, key freshness and key generation. In typical scenarios, nodes are solely responsible for the generation, secure storage and use of their private key often without the necessary entropy required. That can lead to vulnerabilities that can expose private key information to an adversary, tampering the user's BC account and difficulties related to tracking criminal behaviour against modified BC information. The authors have not investigated crypto-viral extortion software attacks as this is considered outside the scope of this present work.

Malicious contracts can be executed by skilful adversaries for malicious activities that can exploit security vulnerabilities in our four legitimate smart contracts [57], [58]. Significant risks apply when external contracts are called without controlling their flow, allowing changes to the data in an unexpected way. Irregularities and exception handling, privacy values leaked by nodes and orderers and altering contracts during or after deployment are typical attack vectors potentially present in our network. We accept that risk as low in our Cydon Platform as both the searching and linking smart contracts are strictly executed via an authorisation and authentication module as part of our work. BC support interoperation between different users, applications, and processes. The way that data

is stored, processed and updated create a substantial economic benefit and motivation for adversaries to interfere with the security management and self-organisation of the platform. Certain elements related to privacy enhancement as part of the BC operation might render it impossible to verify and trace users true identity. We have partially mitigated this threat by both the creation of the Cydon token and the secure and decentralised storage of data across multiple filesystems. There are possibilities of misconfigured transactions, increasing the complexity of the script that controls our contracts' injection. These contract-type transactions must be verified, and the accuracy of the script that controls them should be tested. That verification process becomes a necessity, especially in cases that the BC network grows more complex and transaction facilitators are scattered. This is something that we have documented as part of our future work.

## V. Cydon Testbed and Performance analysis

The structure of the test environment is shown in Fig. 3. To better emulate a real-world scenario, the network is divided into several virtual machines (VMs) administrated by VirtualBox. Each virtual machine includes many Docker containers interconnected among them and further connections are established to other devices through a Docker Swarm network. The typical network consists of a single VM containing the application frontend, a VM for the storage and a configurable number of VMs representing the organisations accessing the BC. The application frontend is a container built around a web application written in ASP.NET Core that exposes all the platform's functionalities through standard API, easily accessible from any client.

A web-based client has been developed in WebAssembly / Blazor which is able to provide high performance as the code is compiled and natively executed within the browser itself. The frontend machine also incorporates the BC orderer and Hyperledger Explorer, a tool which provides a quick analysis of BC structure through specific APIs and dashboards (See Fig. 4a).

Admin users can access the administration area to activate, lock or delete users and groups via the Frontend docker container. Users have specific permissions, whenever they upload new content, search it or both making them the fundamental recipients associated to each upload (see Fig. 4b).

On the other hand, groups provide an elegant way to allow data access to an entire business unit or office: adding (or removing) a user to (from) a group immediately allows (blocks) him to access all the uploaded content associated with that group, without the need to manually set the corresponding permissions. In the main section of the Frontend container users can upload new content and execute search queries. Files can be uploaded and associated with one or more significant keywords that characterise the data being sent and will help future searches.

Moreover, the user has the possibility to set the allowed recipients, by choosing from specific users and groups. To search for a document, the user needs to provide the cydon token generated during the upload. Furthermore, keywords can be used to limit the results-set further. A dashboard section is designed to summarise the usage of the system providing, among other things, KPIs regarding the number of uploaded files, the top accessed contents and health of the system itself.

The storage VM contains a configurable number of IPFS nodes each one accompanied by the corresponding cluster node necessary to set up the content replica. Inside each organisation's VM, there is the corresponding Certificate Authority for managing PKI, a CLI (Command Line Interface) container to manage the organisation and a configurable number of peer nodes. Each node is flanked by the corresponding CouchDB database in which the transaction-generated world-state is saved. Tests were performed on a server equipped with an i7-5930K CPU consisting of 12 cores at 3.50GHz and 64Gb of memory, allowing us to provide two cores and 8gb to each machine. 15-nodes storage was created together with three VMs corresponding to three organizations. Each VM contains a variable number of peers between 2 and 10. Every peer was able to endorse a transaction and it was injected with the smart contracts, leading to the creation of an additional Docker entity, for a grand total of 200 containers.

The performance of the Cydon platform has been evaluated in all its main functionalities, i.e., the upload of new files, the search for contents given the corresponding token and their download. As a first step, an application called Cydon-Cydon command line interface (CCLI) was developed to allow the invocation of all exposed APIs from the command line, greatly facilitating the subsequent testing phases. The CCLI was developed in Golang and, based on the arguments specified in the command line, is able to talk all the Cydon platform APIs. For example, one can retrieve the list of available tokens or create new ones. Similar functions are available for the storage part, user management and smart contracts invocation. The result of each sub-command has been formatted so as to be immediately reusable as a parameter of a subsequent operation (eg the creation of the token returns the *public Id* that can be used for uploading the file). This composition of commands generates a chain of actions that facilitates the batch execution of all Cydon functions, including our tests and benchmarks.

In order to elicit the performance of our smart contracts, we separate their evaluation from the rest of the processes, especially, the file encoding/decoding performance in both the upload and download phases. To achieve this goal, during our tests, the smart contracts were injected directly into the peers of the BC, exploiting the fact that each one has an independent instance of the chain-code. Results were collected in Matlab by accessing APIs provided by Hyperledger Explorer, in order to extract timestamps and other useful data for the overall performance of Cydon.

With regards to the upload phase, Fig. 5b, 5d show the ability to generate $2550 - 3000$ transactions per minute, corresponding to about $150$ blocks. It is interesting to see how six peers seem to strike an optimal balance between invocations parallelism and consensus complexity, which must collect all responses by the endorser peer. Analogously, Fig. 5a demonstrates how Cydon is capable of performing $500$ to $1000$ searches per minute, based on the number of requesting peers and files associated with the token. The loss of performance

TABLE IV: Potential threats against the Cydon's key security attributes

| Attributes | Threats | |
|---|---|---|
| | Inbound/Outbound; Data in Transit | Stored; Data in resident |
| Confidentiality | Copy/sniff | Access; copy |
| Integrity | Spoof; relay | Edit |
| Availability | Overload; | Delete; overload |
| Authenticity | Spoof; relay | Edit |
| Productivity | Overload | Overload |
| Proprietary | Propagate; relay | Access; copy |

is not proportional to the growth of the number of records, confirming that it is not necessary to go through the whole BC to get results, but to only access the underlying database selectively for that process. Therefore, the search and retrieve overhead should be attributed to the greater size of the response, which contains all the files' anchors. Work in [59] investigates the overhead for compressed data using Cloud as the platform.

The tests show that the number of transactions obtainable with Cydon is at most a few thousand transactions per minute. Compared to relational databases, this is a result of several orders of magnitude smaller than is directly related to the use of a distributed ledger technology and its consensus methods. Where the centralized database searches for maximum performance, the primary purpose of the blockchain is to save data in an unmodifiable and unassailable way. As far as queries are concerned, performances are generally less spaced since the most common blockchain implementations, including Hyperlegder Fabric, maintain the current status in a NoSQL database avoiding the need to cross all the blocks. The limited scalability is another disadvantage of the blockchain compared to traditional approaches: in Cydon the inevitable growth is minimized by saving the files in an off-chain storage but the possibility of using side-chains is being evaluated.

The encoding and decoding part of the files has been evaluated separately, using scripts capable of repeatedly sending data of different sizes. The impact of symmetric encoding is evident in Fig. 6 in the case of reduced file-size. The *Enc time* and *Plain time* series correspond to the average time required (expressed in milliseconds) to add a new file to the storage, encrypted and unencrypted respectively. For each evaluated file size, these times result in an average platform throughput, expressed on the secondary y-axis as files per minute. The impact of data encryption has been estimated as the percentage increase of the time required by the encoded upload compared to the unencrypted one.The encoding process exhibits up to a 50% performance drop compared to the same plaintext content. With the growth in file size, the weight of the encoding decreases by up to 9% for 20MB files. The frontend was developed using the Asp.Net Core 2.0 framework, inheriting its characteristics of speed and modularity. The choice was made considering the expertise of the development team and the fact that the platform is cross-platform and easily integrated with Docker technology, which are critical features for our testbed. However, there are also limitations including the maximum body size of a request, which amounts to 30MB, resulting in a corresponding limit to the files uploaded. However, it was considered that this limitation does not render the tests performed ineffective, as the incidence of encryption is more evident on small files. Furthermore, the impact of bigger files to storage was considered out of the scope of this article, as the current IPFS-based implementation serves as a feature-complete example rather than a high performant solution. In the majority of use cases, such storage is replaced with on-cloud or on-premises solutions, often already available to users.

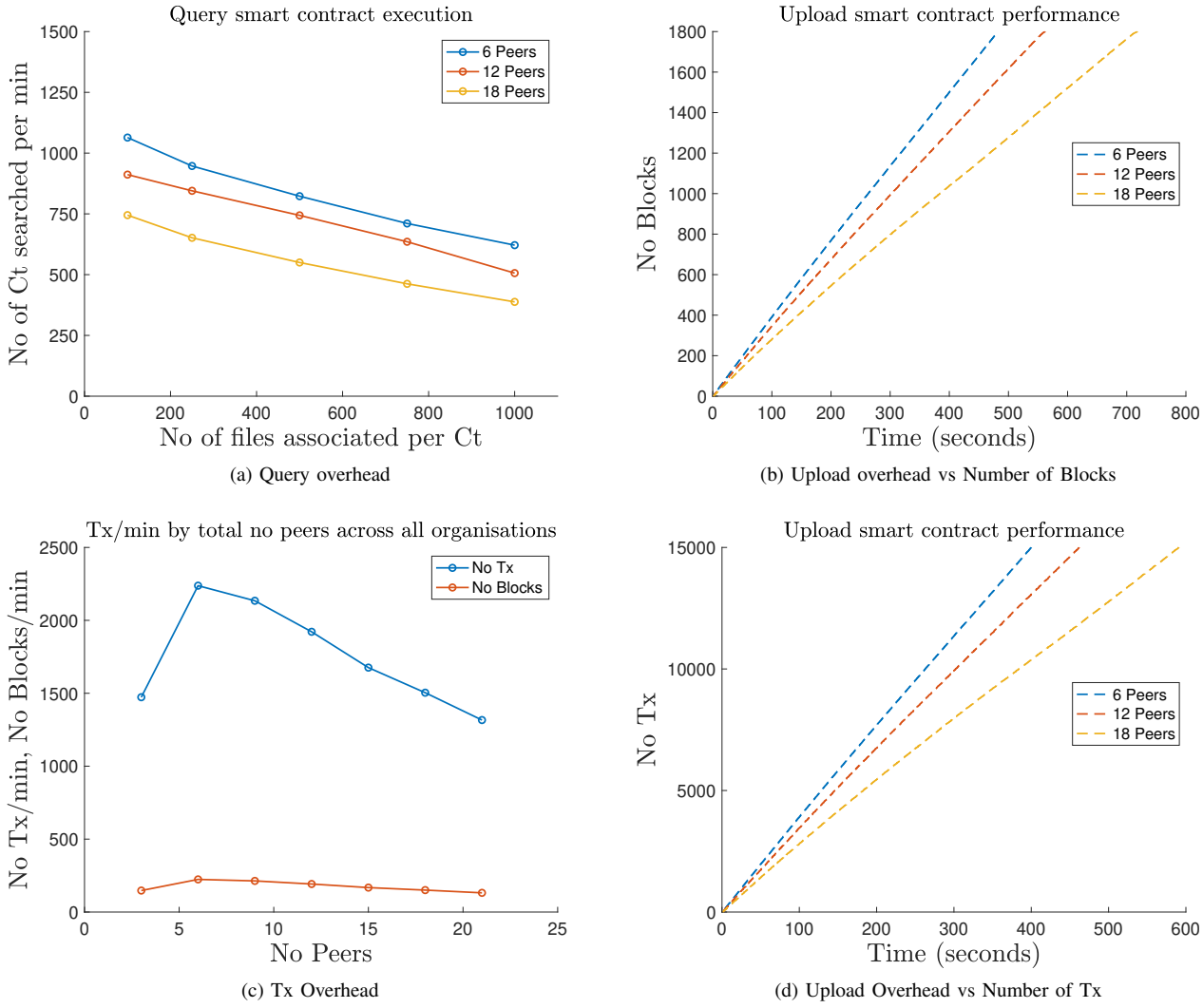
## VI. Conclusion

Special-purpose BCs from healthcare to industrial IoT, have been recently introduced in the public domain. These solutions have been portrayed as a valid alternative to centralised architectures for trusted nodes. BC have the ability to process more transactions and perform decentralised data sharing and processing within and across the supply-chain. Their ability to simplify business cross-border communications and data movement and management in a large scale provides more effective business collaboration and enhance security. In this work, we proposed Cydon, a data management platform that employs a novel crypto-token generation engine and search mechanism which runs over a private permissioned BC. Cydon delivers an always-on audit trail of data flows, while securely distributes and retrieves data from different business entities within a private distributed network.

To the best of our knowledge, Cydon, is the first real-world implementation of a PoC to demonstrate the potentials to leverage BC capabilities to search, locate and retrieve encrypted and distributed data whilst producing a detailed and immutable audit trail for reporting purposes. Part of our future work includes further performance testing and refinements of our algorithm and its associated distributed applications for different BC technologies and consensus mechanisms. We also seek to store Cydon token permissions directly in the BC network instead of the server database. In this way we will also be able to audit changes in our cydon's token attributes for users that their authorisation levels have changed over time.

## Acknowledgments

This work has been funded under the Department of Culture, Media & Sports (DCMS) under the contract no 133728 as part of the Cyber Security Academic Startups Accelerator Programme (ASAP). The authors would also like to thank Mr George Samartzidis for his initial contribution to the software development of the Cydon platform. Parts of this work have been filed under UK patent no 1812770.4.




(a) Query overhead

(b) Upload overhead vs Number of Blocks

(c) Tx Overhead

(d) Upload Overhead vs Number of Tx

Fig. 5: Cydon's overall computational and performance capability

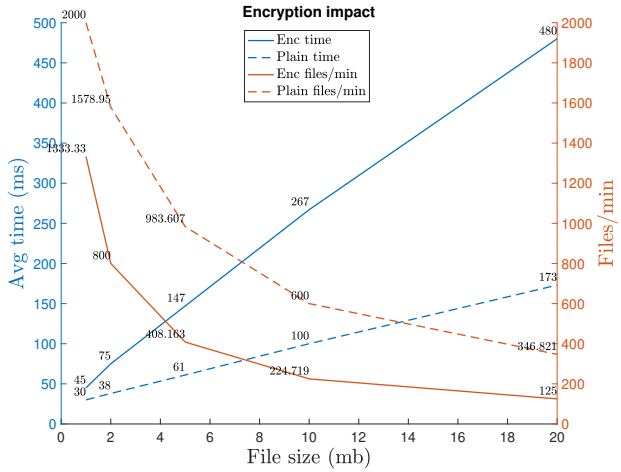



Fig. 6: Cydon's Cryptographic Overhead for different file sizes

## REFERENCES

[1] K.-K. R. Choo, "The cyber threat landscape: Challenges and future research directions," *Computers Security*, vol. 30, no. 8, pp. 719 – 731, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404811001040

[2] S. Tiwari, H. Wee, and Y. Daryanto, "Big data analytics

in supply chain management between 2010 and 2016: Insights to industries," *Computers Industrial Engineering*, vol. 115, pp. 319 – 330, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360835217305508

[3] D. Arunachalam, N. Kumar, and J. P. Kawalek, "Understanding big data analytics capabilities in supply chain management: Unravelling the issues, challenges and implications for practice," *Transportation Research Part E: Logistics and Transportation Review*, vol. 114, pp. 416 – 436, 2018.

[4] K. Wilson and L. Khansa, "Migrating to electronic health record systems: A comparative study between the united states and the united kingdom," *Health Policy*, vol. 122, no. 11, pp. 1232 – 1239, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168851018304214

[5] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for iot data," in *2017 IEEE International Conference on Web Services (ICWS)*, June 2017, pp. 468–475.

[6] R. Fildes, P. Goodwin, and D. Önkal, "Use and misuse of information in supply chain forecasting of promotion effects," *International Journal of Forecasting*, vol. 35, no. 1, pp. 144 – 156, 2019, special Section: Supply Chain Forecasting. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207018300062

[7] W. Klibi, A. Martel, and A. Guitouni, "The design of robust value-creating supply chain networks: a critical review," *European Journal of Operational Research*, vol. 203, no. 2, pp. 283–293, 2010.

[8] M. H. Hugos, *Essentials of supply chain management*. John Wiley & Sons, 2018.

[9] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in iot: The challenges, and a way forward," *Journal of Network and Computer Applications*, 2018.

[10] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[11] D. Calvaresi, A. Dubovitskaya, J. P. Calbimonte, K. Taveter, and M. Schumacher, "Multi-agent systems and blockchain: Results from a systematic literature review," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 2018, pp. 110–126.

[12] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, Jan 2018.

[13] D. Calvaresi, A. Dubovitskaya, D. Retaggi, A. F. Dragoni, and M. Schumacher, "Trusted registration, negotiation, and service evaluation in multi-agent systems throughout the blockchain technology," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2018, pp. 56–63.

[14] R. Reisman, "Blockchain serverless public/private key infrastructure for ads-b security, authentication, and privacy," in *AIAA Scitech 2019 Forum*, 2019, p. 2203.

[15] R. M. Parizi and A. Dehghantanha, "On the understanding of gamification in blockchain systems," in *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Aug 2018, pp. 214–219.

[16] P. R. Kleindorfer and G. H. Saad, "Managing disruption risks in supply chains," *Production and Operations Management*, vol. 14, no. 1, pp. 53–68. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1937-5956.2005.tb00009.x

[17] G. Lu, X. Koufteros, and L. Lucianetti, "Supply chain security: A classification of practices and an empirical study of differential effects and complementarity," *IEEE Transactions on Engineering Management*, vol. 64, no. 2, pp. 234–248, May 2017.

[18] L. Xu, L. Chen, Z. Gao, Y. Chang, E. Iakovou, and W. Shi, "Binding the physical and cyber worlds: A blockchain approach for cargo supply chain security enhancement," *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, pp. 1–5, 2018.

[19] H. J. Korpela Kari and D. Tomi, "Digital supply chain transformation toward blockchain integration," 01 2017.

[20] J. Chen, Z. Lv, and H. Song, "Design of personnel big data management system based on blockchain," *Future Generation Computer Systems*, vol. 101, pp. 1122–1129, dec 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X19313354

[21] A. Pazaitis, P. D. Filippi, and V. Kostakis, "Blockchain and value systems in the sharing economy: The illustrative case of backfeed," *Technological Forecasting and Social Change*, vol. 125, pp. 105 – 115, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0040162517307084

[22] E. Bandara, W. K. Ng, K. De Zoysa, N. Fernando, S. Tharaka, P. Maurakirinathan, and N. Jayasuriya, "Mystiko - Blockchain Meets Big Data," *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, pp. 3024–3032, 2019.

[23] D. Rachmawati, J. T. Tarigan, and A. B. C. Ginting, "A comparative study of Message Digest 5(MD5) and SHA256 algorithm," *Journal of Physics: Conference Series*, vol. 978, p. 012116, mar 2018. [Online]. Available: http://stacks.iop.org/1742-6596/978/i=1/a=012116?key=crossref.efe7b54993adc5b69ef281e39faae168

[24] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 1085–1100.

[25] H. T. Vo, A. Kundu, and M. K. Mohania, "Research directions in blockchain data management and analytics." in *EDBT*, 2018, pp. 445–448.

[26] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, July 2018.

[27] D. Berbecaru and L. Albertalli, "On the performance and use of a space-efficient merkle tree traversal algorithm in real-time applications for wireless and sensor networks," in *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct 2008, pp. 234–240.

[28] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*. IEEE, Jun. 2017, pp. 557–564.

[29] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: ACM, 2018, pp. 30:1–30:15.

[30] H. Kim and M. Laskowski, "A perspective on blockchain smart contracts: Reducing uncertainty and complexity in value exchange," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, July 2017, pp. 1–6.

[31] A. Imeri, D. Khadraoui, and N. Agoulmine, *Blockchain Technology for the Improvement of SCM and Logistics Services: A Survey*. Springer, 2019.

[32] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao, "A survey of blockchain applications in different domains," in *Proceedings of the 2018 International Conference on Blockchain Technology and Application*. ACM, 2018, pp. 17–21.

[33] D. Lee and R. H. Deng, "Handbook of blockchain, digital finance, and inclusion: Cryptocurrency, fintech, insurtech, and regulation," 2018.

[34] Y. Ejiri, E. Ikeda, and H. Sasaki, "Realization of data exchange and utilization society by blockchain and data jacket: Merit of consortium to accelerate co-creation," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 180–182.

[35] H. G. Do and W. K. Ng, "Blockchain-based system for secure data storage with private keyword search," in *2017 IEEE World Congress on Services (SERVICES)*, June 2017, pp. 90–93.

[36] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 17–30. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978389

[37] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *Proceedings of the 25th USENIX Conference on Security Symposium*, ser. SEC'16. Berkeley, CA, USA: USENIX Association, 2016, pp. 279–296. [Online]. Available: http://dl.acm.org/citation.cfm?id=3241094.3241117

[38] Z. Wang, Y. Tian, and J. Zhu, "Data sharing and tracing scheme based on blockchain," in *2018 8th International Conference on Logistics, Informatics and Service Sciences (LISS)*. IEEE, 2018, pp. 1–6.

[39] T. Renner, J. Müller, and O. Kao, "Endolith: A blockchain-based framework to enhance data retention in cloud storages," in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, 2018, pp. 627–634.

[40] I. Zikratov, A. Kuzmin, V. Akimenko, V. Niculichev, and L. Yalansky, "Ensuring data integrity using blockchain technology," in *2017 20th*

*Conference of Open Innovations Association (FRUCT)*, April 2017, pp. 534–539.
[41] M. Kumar, A. K. Singh, and T. V. Suresh Kumar, "Secure log storage using blockchain and cloud infrastructure," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, July 2018, pp. 1–4.
[42] L. Zhu, Y. Wu, K. Gai, and K.-K. R. Choo, "Controllable and trustworthy blockchain-based cloud data management," *Future Generation Computer Systems*, vol. 91, pp. 527 – 535, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18311993
[43] N. Kaaniche and M. Laurent, "Bdua: Blockchain-based data usage auditing," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, July 2018, pp. 630–637.
[44] I. Khanna, "Drug discovery in pharmaceutical industry: productivity challenges and trends," *Drug Discovery Today*, vol. 17, no. 19, pp. 1088 – 1102, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359644612001833",
[45] H. Jahankhani and S. Kendzierskyj, "Digital transformation of healthcare," in *blockchain and Clinical Trial*, A. J. G. E. Hamid Jahankhani, Stefan Kendzierskyj and H. Al-Khateeb, Eds. USA: Springer, 2019, ch. 1, pp. 31–53.
[46] H. D. Gregory Epiphaniou and H. Al-Khateeb, "Blockchain and healthcare," in *blockchain and Clinical Trial*, A. J. G. E. Hamid Jahankhani, Stefan Kendzierskyj and H. Al-Khateeb, Eds. USA: Springer, 2019, ch. 1, pp. 1–31.
[47] G. E. Haider Al-Khateeb and H. Daly, "Blockchain for modern digital forensics: The chain-of-custody as a distributed ledger," in *blockchain and Clinical Trial*, A. J. G. E. Hamid Jahankhani, Stefan Kendzierskyj and H. Al-Khateeb, Eds. USA: Springer, 2019, ch. 1, pp. 149–169.
[48] S. Kalkman, M. Mostert, C. Gerlinger, J. J. M. van Delden, and G. J. M. W. van Thiel, "Responsible data sharing in international health research: a systematic review of principles and norms," *BMC Medical Ethics*, vol. 20, no. 1, p. 21, Mar 2019. [Online]. Available: https://doi.org/10.1186/s12910-019-0359-9
[49] J. Li, J. Wu, and L. Chen, "Block-secure: Blockchain based scheme for secure p2p cloud storage," *Information Sciences*, vol. 465, pp. 219–231, 2018.
[50] C. Yang, X. Chen, and Y. Xiang, "Blockchain-based publicly verifiable data deletion scheme for cloud storage," *Journal of Network and Computer Applications*, vol. 103, pp. 185 – 193, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517303910
[51] M. J. M. Chowdhury, A. Colman, M. A. Kabir, J. Han, and P. Sarda, "Blockchain as a notarization service for data sharing with personal data store," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 1330–1335.
[52] P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands, "Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption," *Journal of the American Medical Informatics Association*, vol. 13, no. 2, pp. 121–126, 2006.
[53] Y.-A. de Montjoye, S. S. Wang, A. Pentland, D. T. T. Anh, A. Datta *et al.*, "On the trusted use of large-scale personal data." *IEEE Data Eng. Bull.*, vol. 35, no. 4, pp. 5–8, 2012.
[54] IPFS documentation. (Accessed: 2019-09-22). [Online]. Available: https://docs.ipfs.io
[55] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Bünzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 67–82. [Online]. Available: http://doi.acm.org/10.1145/3243734.3243780
[56] I. E. T. Force. JSON Specification. (Accessed: 2019-09-22). [Online]. Available: http://json-schema.org/specification.html
[57] R. M. Parizi, A. Dehghantanha, K.-K. R. Choo, and A. Singh, "Empirical vulnerability analysis of automated smart contracts security testing on blockchains," in *Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering*, ser. CASCON '18. Riverton, NJ, USA: IBM Corp., 2018, pp. 103–113. [Online]. Available: http://dl.acm.org/citation.cfm?id=3291291.3291303
[58] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352864818301536
[59] M. Aldwairi, A. Hamzah, and M. Jarrah, "Multiplzw: A novel multiple pattern matching search in lzw-compressed data," *Computer Communications*, vol. 145, 06 2019.

**Gregory Epiphaniou** is an Associate Professor in Cybersecurity and Deputy director of the WCRI at the University of Wolverhampton. He has taught in many Universities both nationally and internationally a variety of areas related to Cybersecurity with over 60 international publications in journals, conference proceedings and author in several book chapters. He holds several industry certifications around Information Security, and worked with several government agencies including the MoD in Cybersecurity related projects. He currently holds a subject matter expert panel position in the Chartered Institute for Securities and Investments and acts as a technical committee member for several scientific conferences in Information and network security.

**Prashant Pillai** has over 15 years of research experience and specializes in the area of Communication protocols and Cyber Security. He is currently the Director of the Wolverhampton Cyber Research Institute (WCRI) which consists of 24 academics. His main areas of work are in computer and communication networking (4G/5G, WLAN, WiMAX, satellite, etc.) looking into protocol design and development, radio resource management, mobility management, and various security aspects like design of security architectures and protocols, cryptographic mechanisms, security risk assessment and modelling, security management, secure testing, etc.

**Mirko Bottarelli** was born in 1980 in Milan, Italy. He received his Bachelor and Master's degrees in Computer Science from Università degli Studi di Milano Bicocca, Milan, Italy in 2004 and 2006, respectively. He is currently pursuing a PhD degree in the Faculty of Science and Engineering at the University of Wolverhampton, UK. His research interests are in the area of wireless communication, information theory and physical layer security.

**Haider Al-Khateeb** received the B.Sc. (Hons.) degree in computer science and the Ph.D. degree in cyber security. He was a Lecturer with the University of Bedfordshire. He is currently a Senior Lecturer in cyber security with the School of Mathematics and Computer Science, University of Wolverhampton. He conducts research within the Wolverhampton Cyber Research Institute, and is also a consultant, a trainer, and a Fellow of the Higher Education Academy, U.K. He has authored or co-authored numerous professional and peer-reviewed articles on topics, including authentication methods, IoT forensics, cyberstalking, anonymity and steganography. He specializes in cybersecurity, digital forensics and incident response.

**Mohammad Hammoudeh** is currently the Head of the MMU IoT Laboratory and a Senior Lecturer in computer networks and security with the School of Computing, Math and Digital Technology, Manchester Metropolitan University. He has been a researcher and publisher in the field of big sensory data mining and visualization. He is a highly proficient, experienced, and professionally certified cybersecurity professional, specializing in threat analysis, and information and network security management. His research interests include highly decentralized algorithms, communication, and cross-layered solutions to Internet of Things, and wireless sensor networks

**Carsten Maple** is Professor of Cyber Systems Engineering at WMG's Cyber Security Centre (CSC). He is the director of research in Cyber Security working with organisations in key sectors such as manufacturing, healthcare, financial services and the broader public sector to address the challenges presented by today's global cyber environment. He has an international research reputation and extensive experience of institutional strategy development and interacting with external agencies. He has published over 200 peer reviewed papers and is co-author of the UK Security Breach Investigations Report 2010, supported by the Serious Organised Crime Agency and the Police Central e-crime Unit. Professor Maple is a Fellow of the British Computer Society and Vice chair of the Council of Professors and Heads of Computing, UK.